

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-105981

(P2000-105981A)

(43) 公開日 平成12年4月11日 (2000.4.11)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームト* (参考)
G 1 1 B 20/14	3 4 1	G 1 1 B 20/14	3 4 1 A 5 D 0 4 4
H 0 3 M 7/14		H 0 3 M 7/14	B

審査請求 未請求 請求項の数 8 O L (全 14 頁)

(21) 出願番号 特願平10-275358

(22) 出願日 平成10年9月29日 (1998.9.29)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 能弾 長作

神奈川県川崎市幸区小向東芝町1番地 株

式会社東芝研究開発センター内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

Fターム(参考) 5D044 AB03 BC01 BC02 CC04 GL20

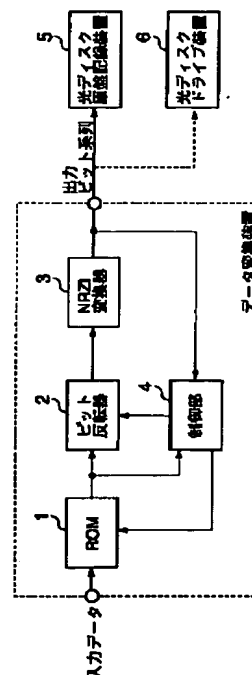
GL21

(54) 【発明の名称】 データ変換方法および装置

(57) 【要約】

【課題】 クロック周波数を高くすることなく、また余分な調整ビットを挿入することなしに、出力ビット系列の直流及び低周波成分を抑圧できるデータ変換方法を提供する。

【解決手段】 ビット“1”に挟まれたビット“0”の連続個数、先頭側のビット“0”の連続個数、後尾側のビット“0”の連続個数が制限されたコード群を有するテーブルを格納したROM 1によりmビットの入力データをnビットのコードに変換し、さらに最後尾のビットを反転するビット反転器2を経て得られたコードをNRZ I変換器3に入力し、制御部4によりNRZ Iパターンの“0”を“-1”、“1”を“+1”として累積加算値(DSV)を求め、コードの最後尾のビットについて反転可能位置におけるビット反転器2の反転実行の有無を次の反転可能位置までのDSVの絶対値がより小さくなるように制御する。



1

## 【特許請求の範囲】

【請求項1】 mビットの入力データをビット“1”と“1”との間のビット“0”の連続個数がd個以上、k個以下、先頭側のビット“0”の連続個数がk1個以下、後尾側のビット“0”の連続個数がk2個以下にそれぞれ制限されたコード群を有する第1のテーブルを用いてnビットのコードに変換するコード変換ステップと、

前記コード変換ステップにより得られたコードの最後尾のビットを適応的に反転するビット反転ステップと、 10

前記コード変換ステップおよびビット反転ステップを経て得られたコードを該コードのビット“1”で出力を反転し、ビット“0”で出力を保持するNRZIパターンに変換するNRZI変換ステップと、

前記NRZIパターンの“0”を“-1”、“1”を“+1”として累積加算値を求める累積加算ステップと、

前記コード変換ステップにより得られたコードのうち後尾側のビット“0”の連続個数が(k-k1)個より大きい現コードに対して、 20

(a) 前記現コードの後尾側のビット“0”の連続個数と次コードの先頭側のビット“0”の連続個数との和がk個を越える場合は前記ビット反転ステップを実行し、

(b) 前記次コードの先頭側のビット“0”の連続個数がd個より小さい場合は前記ビット反転ステップを不実行とし、

(c) 前記(a)(b)の条件に該当しない場合は前記コード変換ステップにより得られたコードの最後尾のビットの少なくとも一部を反転可能位置と判断し、該反転可能位置における前記ビット反転ステップの実行の有無 30  
を次の反転可能位置までの前記累積加算値の絶対値がより小さくなるように制御する反転制御ステップとを有することを特徴とするデータ変換方法。

【請求項2】 前記コード変換ステップは、前記第1のテーブルを用いて変換されたコードのうち、現コードの後尾側のビット“0”の連続個数と次コードの先頭側のビット“0”の連続個数の和がd個より小さい場合には、前記現コードとして、前記第1のテーブルのコード群と同一の条件を満たし、かつ該第1のテーブルのコード群には含まれず、先頭側のビット“0”の連続個数が前記 40  
第1のテーブルを用いて変換されたコードのそれと等しいコード群を有する第2のテーブルにより前記入力データをコードに変換し、

次コードとして、ビット“1”とビット“1”との間のビット“0”の連続個数がd個以上、k個以下で、かつ先頭側のビット“0”の連続個数がd個以上、(k-d+1)個以下、後尾側のビット“0”の連続個数がd個以上、k2個以下にそれぞれ制限されたコード群を有する第3のテーブルにより前記入力データをコードに変換することを特徴とする請求項1記載のデータ変換方法。 50

2

## 【請求項3】 前記反転制御ステップは、

前記第3のテーブルを用いて変換されたコードの後尾側のビット“0”の連続個数がd個より大きいコードに対して、

(d) 後尾側のビット“0”の連続個数と前記第1のテーブルを用いて変換された次コードの先頭側のビット“0”の連続個数の和がk個を越える場合は前記ビット反転ステップを実行し、

(e) 次コードの先頭側のビット“0”の連続個数がd個より小さい場合は前記ビット反転ステップを不実行とし、

(f) 前記(d)(e)の条件に該当しない場合には前記コード変換ステップにより得られたコードの最後尾のビットの少なくとも一部を新たに反転可能位置と判断し、該反転可能位置における前記ビット反転ステップの実行の有無を次の反転可能位置までの前記累積加算値の絶対値がより小さくなるように制御することを特徴とする請求項2記載のデータ変換方法。

【請求項4】 前記mが8、nが12、dが1、kが8、k1が4、k2が8であることを特徴とする請求項1乃至3のいずれか1項記載のデータ変換方法。

【請求項5】 前記コード変換ステップは、周期的に同期コードを挿入する過程を有することを特徴とする請求項1乃至4のいずれか1項記載のデータ変換方法。

【請求項6】 前記反転制御ステップは、前記反転可能位置から前記同期コードが挿入される直前までの前記累積加算値の絶対値がより小さくなるように、該反転可能位置における前記ビット反転ステップの実行の有無を制御するものであることを特徴とする請求項5記載のデータ変換方法。

【請求項7】 前記同期コードが挿入された直後の前記反転可能位置のまでの前記累積加算値の絶対値がより小さくなるように、該同期コードにおけるビット“1”が偶数個のパターンと奇数個のパターンのいずれかを選択する同期コードパターン選択ステップをさらに有することを特徴とする請求項5または6のいずれか1項記載のデータ変換方法。

【請求項8】 mビットの入力データをビット“1”と“1”との間のビット“0”の連続個数、先頭側のビット“0”の連続個数、後尾側のビット“0”の連続個数がそれぞれ制限されたコード群を有するテーブルを用いてnビットのコードに変換するコード変換手段と、前記コード変換ステップにより得られたコードの最後尾のビットを適応的に反転するビット反転手段と、前記コード変換手段およびビット反転手段により得られたコードを該コードのビット“1”で出力を反転し、ビット“0”で出力を保持するNRZIパターンに変換するNRZI変換手段と、

前記NRZIパターンの“0”を“-1”、“1”を“+1”として累積加算値を求める累積加算手段と、

## 3

前記コード変換手段により得られたコードの最後尾のビットのうち少なくとも一部を反転してもしなくとも構わない反転可能位置と判断し、該反転可能位置における前記ビット反転手段による反転の実行の有無を次の反転可能位置までの前記累積加算値の絶対値がより小さくなるように制御する制御手段とを有することを特徴とするデータ変換装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、光ディスクのよう<sup>10</sup>な記録媒体を用いてデジタルデータの記録や再生を行う装置において、デジタルデータを記録媒体上に記録されるビット系列に変換する場合に好適なデータ変換方法および装置に関する。

【0002】

【従来の技術】光ディスクのような記録媒体にデジタルデータを記録する場合、記録ビット系列には様々な特性が求められる。記録ビット系列とは、記録媒体上の記録パターンに対応するビットパターンである。再生専用の光ディスクの場合、記録ビット系列は、反射面の凹凸<sup>20</sup>状のビットとして記録される。記録可能な光ディスクの場合、例えば光磁気ディスクでは、磁化状態の異なるマークとして、また相変化ディスクでは光学定数の異なるマークとして、それぞれ記録ビット系列が記録される。

【0003】従って光ディスクの製造過程、レーザ光で記録ビット系列を読み出す光ヘッドの光学的特性、光ディスクから記録ビット系列を読み出して得られた再生信号を元のデジタルデータに復元する信号処理系の特性などを考慮して、記録すべきデジタルデータ（原データ）を最適な記録ビット系列に変換するデータ変換処理<sup>30</sup>を行う必要がある。

【0004】データがビットとして記録される再生専用の光ディスクの場合、最小ビット長が小さくなると光学的特性の劣化により急激に再生信号の出力が低下するため、最小ビット長は大きい方が望ましい。一方、最大ビット長が大きくなると再生信号の反転回数が減少し、クロックのタイミング再生性能が劣化してジッタが大きくなる結果、符号誤りを生じやすくなるため、最大ビット長は小さい方が望ましい。

【0005】また、光ディスク上にビット等として記録<sup>40</sup>された記録ビット系列の直流及び低周波成分が少ないことも重要である。これらの直流及び低周波成分は、光ディスク上に形成されたトラックを正確にトレースするためのトラッキングサーボに影響を与える場合があり、記録ビット系列を正確に読み出すためには、これらの成分を抑圧する必要がある。

【0006】さらに、検出窓幅が広いことも要求項目として挙げられる。原データを多くのビットに分割して記録を行うと、たとえビット長の条件を満たしていても検出時の時間的位相余裕が小さくなってしまい、同時に再<sup>50</sup>

## 4

生クロック周波数が高くなってしまいうからである。再生クロック周波数が高くなると、信号処理回路の動作速度を高くとる必要があり、回路が高価なものとなってしまいう。

【0007】これらの条件を考慮したデータ変換方式の一つとして、例えば特開平6-284015号公報（以下、文献[1]という）に記載された8/14変換がある。8/14変換は、8ビットのデータを14ビットのコードに変換するデータ変換方式であり、この変換により得られたコードがNRZIパターンのような記録ビット系列に変換された後、記録媒体上に記録される。文献[1]では、記録ビット系列のビット“0”を“-1”、ビット“1”を“+1”として累積加算した値（DSV）が小さくなるように、8ビットから14ビットへのコード変換に用いるテーブルを最適に切り替えているため、記録ビット系列の直流及び低周波成分を十分に抑圧することができる。

【0008】反面、文献[1]ではデータ変換後のビット数が原データの14/8倍に増加してしまうため、検出窓幅が小さくなってしまいうと共に、クロック周波数も同じ比率で上昇してしまうという問題がある。近年、光ディスク装置などのデジタル記録装置に対するデータ転送速度の高速化が要求されているが、データ転送速度を高くするとクロック周波数も増加し、高速で動作できる高価な信号処理回路を必要とする。

【0009】特開昭56-149152号公報（以下、文献[2]という）に、別のデータ変換方式が記載されている。この文献[2]に記載されたデータ変換方式では、原データは1.5倍のビット数のコードに変換される。得られるコードはビット“1”と“1”との間のビット“0”の連続個数が1個以上、7個以下であるため、一般に（1，7）RLL符号と呼ばれる。このデータ変換方式は、比較的低いクロック周波数と規模の小さな回路で実現が可能という特徴を持っているが、文献[1]に記載された8/14変換のようにDSVの管理を一切行っていないため、記録ビット系列の直流及び低周波成分は抑圧されていない。従って、文献[2]の方式ではトラッキング性能に障害を与える場合が発生してくる。

【0010】記録ビット系列の直流及び低周波成分を抑圧するためには、記録すべきデータのビットとは別にDSVを小さくするための調整ビットを挿入すればよいが、このような調整ビットを挿入すると記録媒体の実効的な容量が減少するという新たな問題が生じてくる。また、文献[2]で得られるコードは2ビットを3ビットに変換した場合と4ビットを6ビットに変換した場合が混在する可変長符号のため、ビット誤りが伝搬しやすいといった問題も持っている。

【0011】

【発明が解決しようとする課題】上述したように従来の

## 5

データ変換方式では、トラッキングサーボの安定化等のために記録ビット系列の直流及び低周波成分を抑圧しようとする、クロック周波数が上昇して信号処理回路を構成する上で好ましくなく、また、クロック周波数を低く抑えようとする、記録ビット系列の直流及び低周波成分が抑圧できなかつたり、調整ビットの挿入により記録媒体の実効的な記録容量が減少するといった問題があった。

【0012】本発明は、クロック周波数をあまり高くすることなく、また出力ビット系列に余分な調整ビットを挿入することなしに、出力ビット系列の直流及び低周波成分を抑圧しつつ、入力データをこれと異なるビット数のコードからなる出力ビット系列に変換できるデータ変換方法を提供することを目的とする。

## 【0013】

【課題を解決するための手段】上記の課題を解決するため、本発明においては、まず $m$ ビットの入力データがビット“1”と“1”との間のビット“0”の連続個数が $d$ 個以上、 $k$ 個以下、先頭側のビット“0”の連続個数が $k_1$ 個以下、後尾側のビット“0”の連続個数が $k_2$ 個以下にそれぞれ制限されたコード群を有する第1のテーブルを用いて $n$ ビットのコードに変換される。このコード変換により得られたコードは、その最後尾のビットが適応的に反転される。これらコード変換とビット反転の処理により、各々のコード単独ではもちろん、コードとコードを接続した場合のコード境界部分においてもビット“0”の連続個数が $d$ 個以上、 $k$ 個以下に制限されたコードが得られる。

【0014】これらコード変換およびビット反転の処理を経て得られたコードは、コードのビット“1”で出力を反転し、ビット“0”で出力を保持するNRZIパターンに変換され、出力ビット系列となる。さらに、NRZIパターンの“0”を“-1”、“1”を“+1”として累積加算値(DSV)が求められる。

【0015】ここで、ビット反転処理は以下のように制御されることにより、コード境界部分のビット“0”の連続個数が $d$ 個以上、 $k$ 個以下に制限されると共に、DSVの絶対値がより小さくなるような制御が行われる。

【0016】すなわち、このビット反転の制御に際しては、コード変換ステップにより得られたコードのうち後尾側のビット“0”の連続個数が $(k - k_1)$ 個より大きい現コードに対して、(a)現コードの後尾側のビット“0”の連続個数と次コードの先頭側のビット“0”の連続個数との和が $k$ 個を越える場合はビット反転を実行し、(b)次コードの先頭側のビット“0”の連続個数が $d$ 個より小さい場合はビット反転を不実行とし、さらに(c)として(a)(b)の条件に該当しない場合はコード変換により得られたコードの最後尾のビットの少なくとも一部を反転可能位置と判断し、この反転可能位置におけるビット反転の実行の有無を次の反転可能位置

## 6

置までのDSVの絶対値がより小さくなるように制御する。

【0017】このような本発明によるデータ変換では、例えば $m$ を8、 $n$ を12、 $d$ を1、 $k$ を8、 $k_1$ を4、 $k_2$ を8とすることにより、クロック周波数をあまり高くしたり、また出力ビット系列に余分な調整ビットを挿入することなく、DSVの絶対値をより小さくする制御によって出力ビット系列の直流及び低周波成分を抑圧することが可能となる。

【0018】本発明においては、コード変換に際して第1のテーブルを用いて変換されたコードのうち、現コードの後尾側のビット“0”の連続個数と次コードの先頭側のビット“0”の連続個数の和が $d$ 個より小さい場合には、現コードとして、第1のテーブルのコード群と同一の条件を満たし、かつ第1のテーブルのコード群には含まれず、先頭側のビット“0”の連続個数が第1のテーブルを用いて変換されたコードのそれと等しいコード群を有する第2のテーブルにより入力データをコードに変換し、次コードとして、ビット“1”とビット“1”との間のビット“0”の連続個数が $d$ 個以上、 $k$ 個以下で、かつ先頭側のビット“0”の連続個数が $d$ 個以上、 $(k - \alpha + 1)$ 個以下、後尾側のビット“0”の連続個数が $d$ 個以上、 $k_2$ 個以下にそれぞれ制限されたコード群を有する第3のテーブルにより入力データをコードに変換することがより望ましい。

【0019】このようにすると、例えば $m=8$ 、 $n=12$ 、 $d=1$ 、 $k=8$ というような条件においても、入力データの全てのパターンに対して、コード単独およびコードとコードを接続した場合のコード境界部分においてビット“0”の連続個数が $d$ 個以上、 $k$ 個以下に制限されたコードが得られる。

【0020】また、このように第2のテーブルを併用してコード変換を行った場合には、第3のテーブルを用いて変換されたコードの後尾側のビット“0”の連続個数が $d$ 個より大きいコードに対して、(d)後尾側のビット“0”の連続個数と第1のテーブルを用いて変換された次コードの先頭側のビット“0”の連続個数の和が $k$ 個を越える場合はビット反転を実行し、(e)次コードの先頭側のビット“0”の連続個数が $d$ 個より小さい場合はビット反転を不実行とし、(f)として、(d)

(e)の条件に該当しない場合にはコード変換により得られたコードの最後尾のビットの少なくとも一部を新たに反転可能位置と判断し、この反転可能位置におけるビット反転の実行の有無を次の反転可能位置までのDSVの絶対値がより小さくなるように制御するようにしてもよい。

【0021】このように反転可能位置をさらに多くとり、DSVの管理に利用することによって、出力ビット系列の直流及び低周波成分をより効果的に抑圧することが可能となる。

【0022】さらに、コード変換に際して、周期的に同期コードを挿入する過程を設け、反転可能位置から同期コードが挿入される直前までのDSVの絶対値がより小さくなるように、該反転可能位置におけるビット反転ステップの実行の有無を制御するようにしてもよい。さらに、同期コードが挿入された直後の反転可能位置のまでのDSV値の絶対値がより小さくなるように、該同期コードにおけるビット“1”が偶数個のパターンと奇数個のパターンのいずれかを選択する同期コードパターン選択ステップをさらに設けるようにしてもよい。このよう<sup>10</sup>にすることによって、出力信号系列の直流及び低周波成分をより一層効果的に抑圧することが可能となる。

【0023】また、本発明に係るデータ変換装置は、mビットの入力データをビット“1”と“1”との間のビット“0”の連続個数、先頭側のビット“0”の連続個数、後尾側のビット“0”の連続個数がそれぞれ制限されたコード群を有するテーブルを用いてnビットのコードに変換するコード変換手段と、このコード変換ステップにより得られたコードの最後尾のビットを適応的に反転するビット反転手段と、これらコード変換手段および<sup>20</sup>ビット反転手段により得られたコードを該コードのビット“1”で出力を反転し、ビット“0”で出力を保持するNRZIパターンに変換するNRZI変換手段と、このNRZIパターンの“0”を“-1”、“1”を“+1”として累積加算値(DSV)を求める累積加算手段と、コード変換手段により得られたコードの最後尾のビットの少なくとも一部を反転してもしなくとも構わない反転可能位置と判断し、この反転可能位置におけるビット反転手段による反転の実行の有無を次の反転可能位置までのDSVの絶対値がより小さくなるように制御する<sup>30</sup>制御手段とを有することを特徴とする。

#### 【0024】

【発明の実施の形態】以下、本発明のデータ変換方法をm=8ビットのデジタルデータ(入力データ)をn=12ビットのコードに変換する8/12変換過程を含み、入力データを光ディスクなどの記録媒体への記録に適したビット系列(出力ビット系列)に変換する場合の実施形態について詳細に説明する。

【0025】入力データは、8ビットで表現される256通りのビットパターンを有する。一方、8/12変換<sup>40</sup>後のコードは12ビットで表現される4096通りのビットパターンが存在する。この場合のデータ変換の主たる過程は、これら256通りのパターンと4096通りのパターン間での割り当てである。

【0026】このようなデータ変換に求められる主な条件を再度整理すると、

- (1) 最小ビット長が大きいこと
- (2) 最大ビット長が小さいこと
- (3) 検出窓幅が広いこと
- (4) 再生クロック周波数が低いこと

(5) 出力ビット系列の直流及び低周波成分が少ないこと  
等が挙げられる。

【0027】上記(1)と(2)の条件を満たすために、本実施形態では8/12変換によって得られるコードのビットパターンとして、ビット“0”の連続個数(より正確には、ビット“1”と“1”との間のビット“0”の連続個数)RL0(図7参照)をd=1個以上、k=8個以下に制限した(1, 8)RL符号を用いる。この場合、コードとコードが接続されたときのコードの境界部分においても(1, 8)RL符号の条件を満たすようにすることが必要であり、このために本実施形態では後述するようにビット反転処理を用いる。

【0028】また、上記(3)と(4)の条件を満たすために、本実施形態では上述のように8/12変換を用いる。8/12変換によると、再生クロック周波数は入力データの基本クロック周波数の12/8=1.5倍となり、文献[1]に記載された8/14変換に比較して低く抑えられる。

【0029】さらに、上記(5)の条件を満たすために、本実施形態は上述のビット反転を実行しても実行しなくともよいビットの位置(反転可能位置)のビット反転の実行の有無を制御することで、DSVの管理を行う。

【0030】((1, 8)RL符号について)まず、(1, 8)RL符号について説明する。図1は、(1, 8)RL符号の条件を満たす12ビットパターンの一例を示している。(1, 8)RL符号の条件を満たす12ビットパターンのパターン数は全部で365通り存在し、8ビットよりなる入力データのパターン数の256通りを越えているので、8ビットの入力データを単独で(1, 8)RL符号の条件を満たす12ビットのコードに変換することは十分に可能である。しかし、これら12ビットのコードが連続して接続された場合には、コードの境界部分において(1, 8)RL符号の条件を満たされなくなることがある。

【0031】そこで、本実施形態では以下のようにして変換されたコードの境界部分においても(1, 8)RL符号の条件を満たすデータ変換を行う。このデータ変換の処理は、大きく分けて(i)コード変換処理、(ii)ビット反転処理、(iii)NRZI変換、および(iv)ビット反転制御からなる。以下、これらの各処理について順次説明する。

【0032】(コード変換処理について…その1)図1のビットパターンからなる任意の2つのコードを接続すると、コードの境界部分でビット“1”が連続したり、ビット“0”が最大16個連続する場合が存在することは容易に分かる。そこで、図1のビットパターンから、さらにコードの両端におけるビットパターンに以下の制約を加える。コードの先頭側のビット“0”の連続個数

RL1 (図7参照) を  $k1 = 4$  個以下、後尾側のビット“0”の連続個数RL2 (図7参照) を  $d = 1$  個以上、 $k2 = 8$  個以下とする。これらのコードを主コードAとして割り当てる。なお、この制約を付加することによって、取り得るコード数は207通りに減少する。図2に、主コードAに割り当て可能なビットパターンを示す。これは第1のテーブルの内容を表しており、このようなビットパターンを持つ主コードAからなるコード群を入力データの種々のビットパターンに対応させて記述している。コード変換処理では、基本的にこの第1のテ

10  
 \*  
 RL3 = 9の場合 (8 : 1), (7 : 2), (6 : 3), (5 : 4)  
 RL3 = 10の場合 (8 : 2), (7 : 3), (6 : 4)  
 RL3 = 11の場合 (8 : 3), (7 : 4)  
 RL3 = 12の場合 (8 : 4)

これから理解されるように、現コードの後尾側におけるビット“0”の連続個数RL2が  $(k - k1) = 4$  個より大きく (5個以上)、次コードの先頭側におけるビッ

20  
 ト“0”の連続個数RL1が  $d = 1$  個以上の場合に、現コードと次コードの境界部分におけるビット“0”の連続個数RL3が8個を超える可能性がある。  
 【0035】そこで、後尾側におけるビット“0”の連続個数RL2が  $(k - k1) = 4$  個を超える27通りのパターンのコードに対して、図3に示すようにコードの最後尾ビットをビット反転して“1”としたパターンを「反転コード」として割り当てる。コード境界部分におけるビット“0”の連続個数RL3に応じてビット反転の実行の有無を適宜選択することにより、(1, 8) R

30  
 LL符号の条件を常に満足させることができる。図2に示した主コードAに属する207通りのパターンの最後尾ビットは必ず“0”であるため、ビット反転処理後の反転コードは主コードAとは重複しない。  
 【0036】(コード変換処理について…その2)ところで、前述した主コードAからなるコード群を有する第1のテーブルのみでは、最大207通りのパターンの入力データにしか対応できない。8ビットで表現可能な256通りのパターンの入力データ全てに対してコードを割り当てるためには、残る49通りのパターンの入力デ

40  
 ータに対して別のパターンをコードとして割り当てる必要がある。  
 【0037】本実施形態では、これら49通りのパターンの入力データに対して、新たに最後尾ビットが“1”のパターンをコードとして追加する。(1, 8) RLL符号の条件を満たし、かつ先頭側におけるビット“0”の連続個数RL1が  $k1 = 4$  個以下で、最後尾ビットが“1”のパターンは、全部で129通り存在する。このような条件を満たすパターンのうち、図3の反転コードとして使用した27通りのパターンを除く102通りの

\*【0033】(ビット反転処理について…その1) 上述した主コードAからなるコード群を有する第1のテーブルのみを使用してデータ変換を行っても、コード境界部分においてビット“0”の連続個数 (現コードの後尾側のビット“0”の連続個数と次コードの先頭側のビット“0”の連続個数との和) RL3 (図7参照) が  $k = 8$  個を超える場合があり得る。コード両端のパターンの制約から、コード境界部分におけるビット“0”の連続個数RL3が  $k = 8$  を越えるケースは、現コードの後尾側のビット“0”の連続個数RL2と次コードの先頭側のビット“0”の連続個数RL1の組み合わせ (RL2, RL1) が以下の10通りの場合に発生する。

【0034】

パターンのみが実際に使用可能であり、これらのパターンのコードを主コードBとして割り当てる。

【0038】図4に、主コードBに割り当て可能なビットパターンを示す。これは前述した第2のテーブルの内容を表しており、このようなビットパターンを持つ主コードBからなるコード群を入力データの種々のビットパターンに対応させて記述している。

【0039】ところで、このような最後尾ビットが“1”の主コードBを割り当てると、コードの境界部分でビット“1”が連続し、(1, 8) RLL符号の条件を満たせなくなる場合が発生する。これを避けるため、次コードの先頭ビットが“0”か“1”かに応じて、主コードBから2種類のコード (特定コードと通常コード) のいずれかを選択して現コードに割り当てると共に、次コードには後述する第3のテーブルから得られる「代替コード」を割り当てることにする。

【0040】例えば、主コードBを後尾側の3ビットが特定のパターン、例えば“101”となる「特定コード」と、その他の「通常コード」とに分類して、両者の組み合わせを準備する。そして、次コードがビット“1”で始まる場合には特定コードを使用し、“0”で始まる場合には通常コードを使用する。

【0041】一つのデータに対応する特定コードと通常コードの組み合わせは、前述のビット反転処理の実行の有無の判定を容易にするために、2つのコードの先頭側のビット“0”の連続個数が等しくなるように割り当てると良い。その結果、主コードBから50通りのデータに対するコード割り当てができる。図5に、この方法により主コードBから選択した特定コードと通常コードの組み合わせを示す。

【0042】上述した207通りの主コードAと50通りの主コードBを組み合わせると、合計257通りのコードを用意できるため、8ビットで表現される256通

11

りのパターンの入力データに対するコードの割り当ては十分可能といえる。

【0043】次に、コード境界部分においてビット

“1”が連続する場合に使用する代替コードについて説明する。代替コードは、現コードの後尾側の3ビットが特定コードである“101”の場合にのみ、次コードとして使用する。従って、代替コードには主コードAおよびBとして使用したコードと重複するパターンも使用することができる。

【0044】代替コードに要求される条件は、先頭側に 10  
おけるビット“0”の連続個数RL1が $d=1$ 個以上、 $k-\alpha+1=8$ 個以下、後尾側におけるビット“0”の連続個数RL2が $d=1$ 個以上、 $k_2=8$ 個以下で、かつ、いずれも(1, 8)RLL符号の条件(ビット“1”と“1”との間のビット“0”の連続個数RL0が $d=1$ 個以上、 $k=8$ 個以下)を満たすことである。この条件を満足するパターンは、139通り存在する。

【0045】図6に、代替コードに割り当て可能なビットパターンの一例を示す。これは第3のテーブルの内容を表しており、このようなビットパターンを持つ代替コ 20  
ードからなるコード群を入力データの特定のビットパターンに対応させて記述している。このような代替コードの割り当てが必要となる入力データは、図2の第1のテーブルに記述された主コードAとして使用されるコード群と、図5の第2のテーブルに記述された主コードBとして使用されるコード群のうち、ビット“1”で始まるコードに対応するパターンに限られる。図2および図5において、ビット“1”で始まるコードは107通り存在し、代替コードとして取り得るパターン数よりも少ない。よって、代替コードの割り当ては十分に可能であ 30  
る。

【0046】(ビット反転処理について…その2)ところで、代替コードに対しても主コードAの場合と同様に、後尾側におけるビット“0”の連続個数RL2が $(k-k_1)=4$ 個を越えるコードに対して、次コードとの境界部分におけるビット“0”の連続個数RL3が $k=8$ 個を越える場合がある。その場合には、代替コードの最後尾ビットをビット“1”に反転させる。後尾側におけるビット“0”の連続個数RL2が2個以上の代替コードに対して、反転コードを割り当てることができ 40  
る。

【0047】なお、最後尾ビットが“1”のパターンのコードは反転前の代替コードには含まれていないため、このビット反転処理によりコードが重複することは無い。また、代替コードの後尾側におけるビット“0”の連続個数RL2は $k=8$ 個以下に制限されているため、

RL3=3の場合 (2:1)

RL3=4の場合 (2:2), (3:1)

RL3=5の場合 (2:3), (3:2), (4:1)

RL3=6の場合 (2:4), (3:3), (4:2), (5:1)

12

\*コード境界部分におけるビット“0”の連続個数RL3が8個を越える場合は、必ず次コードはビット“0”で始まっている。従って、このビット反転処理によりビット“1”が連続することは無い。

【0048】(NRZI変換について) 上述したコード変換処理およびビット反転処理を経て得られたコードはさらにNRZI (Non Return to Zero-Inverse)変換され、最終的な出力ビット系列となる。NRZI変換は、入力されたコードを該コードのビット“1”で出力を反転し、ビット“0”で出力を保持するような二値パターン(NRZIパターン)に変換する処理である。

【0049】(ビット反転制御について) 次に、本実施形態における出力ビット系列の直流及び低周波成分の抑圧方法について説明する。直流及び低周波成分を抑圧するために、NRZIパターンのビット“0”を“−1”、ビット“1”を“+1”として累積加算を行って累積加算値(DSV)を求め、このDSVの絶対値をより小さくするような制御を行うことは公知である。DSVは電気信号に表現すると電荷蓄積量に相当し、信号の直流レベルを示している。

【0050】本実施形態では、前述したビット反転処理において必ずビット反転を実行すべきビット以外に、ビット反転を行っても行わなくとも構わないビット(反転可能位置)が存在することに着目し、現在の反転可能位置から次の反転可能位置までのDSVの絶対値をより小さくするように、現在の反転可能位置の少なくとも一部におけるビット反転の実行の有無を制御する。以下、このビット反転制御について説明する。

【0051】なお、このビット反転制御に関しては、上述した全ての反転可能位置を反転可能位置とせず、その一部のみをビット反転可能位置として、比較的長いデータ区間でDSVの絶対値を算出するようにしてもよい。

【0052】前述したように、本実施形態の8/12変換によるデータ変換方法では、主コードAと代替コードの一部に対して、コード境界部分におけるビット“0”の連続個数RL3が $k=8$ 個を越えないようにするために、前述したビット反転処理を行っている。このビット反転処理に着目すると、ビット反転前と反転後の2種類のコード接続状態のいずれにおいても、コード境界部分におけるビット“0”の連続個数RL3の制約(RL3が $k=8$ 個以下)を満足できる可能性がある。現コードの後尾側におけるビット“0”の連続個数RL2と、次コードの先頭側におけるビット“0”の連続個数RL1との組み合わせ(RL2, RL1)が以下の18通りの場合がこれに該当する。

【0053】

13

RL3=7の場合 (3:4), (4:3), (5:2), (6:1)

RL3=8の場合 (4:4), (5:3), (6:2), (7:1)

反転コードが用意されているコードであって、しかもコード境界部分における(RL2, RL1)の組み合わせが上記18通りのいずれかとなっている場合には、ビット反転処理の実行の有無をビット“0”の連続個数を制限すること以外の用途、すなわちDSVの管理のために制御することができる。

【0054】ここで、(RL2, RL1)の組み合わせがRL3=3の場合の(2:1)、RL3=4の場合の(2:2)、RL3=5の場合の(2:3)、RL3=6の場合の(2:4)については、ビット反転処理を行うと代替コードの後尾側の3ビットが前述した特定コードのそれと同じ“101”になる。従って、例えば特定コードの後尾側3ビットがビットエラーを起こして通常コードと同じになってしまうと、代替コードが特定コードと見なされる結果、代替コードの次のコードが代替コードと誤認されてしまい、データ変換エラーが発生する。

【0055】このような不都合を避けるため、本実施形態では現コードの後尾側におけるビット“0”の連続個数RL2と、次コードの先頭側におけるビット“0”の連続個数RL1との組み合わせ(RL2, RL1)が上に挙げた(2:1), (2:2), (2:3), (2:4)の場合に限っては、代替コードの最後尾ビットに対するビット反転処理を禁止している。なお、このようなエラー耐性を特に考慮する必要がなれば、(RL2, RL1)が(2:1), (2:2), (2:3), (2:4)の場合についても、代替コードの最後尾ビットに対するビット反転処理を行って構わない。

【0056】上述したビット反転制御により、コード変換処理で得られたコードの1ビットのみをビット“0”から“1”に反転すると、NRZI変換後の信号(出力ビット系列)は、反転ビット以降の信号極性が反転する。

【0057】図8に、通常コードおよび反転コードに対応する出力コード、NRZIパターン、出力波形およびDSVの変化を示す。これから分かるように、ビット反転を行うと、以降のDSVの変化方向が逆転する。図8の例では、ビット反転を行わない方がDSVの絶対値を小さくすることができることが分かる。

【0058】従って、ビット反転を行った場合と行わなかった場合に対するDSVをそれぞれ算出し、次にビット反転が可能となる位置において、DSVの絶対値がより小さくなるようにビット反転の実行の有無を制御することが可能となる。このようにビット反転処理を制御することにより、出力信号系列の直流及び低周波成分を効果的に抑圧することができる。

【0059】(本発明のデータ変換方法が適用されるシステムについて) 図9に、上述したデータ変換方法を光

14

ディスク原盤記録装置または光ディスクドライブ装置に供給される記録ビット系列を生成するシステムに適用した例を示す。記録ビット系列生成システムは、ROM(リードオンリーメモリ)1、ビット反転器2、NRZI変換器3および制御部4からなるデータ変換装置によって構成される。

【0060】ROM1には、前述した第1、第2および第3のテーブルが格納されている。データ変換装置に入力された入力データはROM1にアドレスデータとして与えられ、制御部4による制御の下でROM1からコードが読み出される。ROM1から読み出されたコードはビット反転器2に入力され、制御部4による制御の下で最後尾のビットが適応的に反転処理される。ビット反転器2で処理されたコードは、NRZI変換器3によりNRZIパターンに変換され、データ変換装置の出力ビット系列となる。この出力ビット系列は、光ディスク原盤記録装置5または光ディスクドライブ装置6に記録ビット系列として入力される。

【0061】光ディスク原盤記録装置5は、光ディスクを製造するための原盤にビットとしてデータを記録する装置であり、記録ビット系列に従って露光用光ビームの強度変調を行い、原盤に形成されたフォトレジスト層を露光する。次に、露光後のフォトレジスト層を現像することにより、原盤に記録ビット系列に対応したビット列を形成する。このような一連の工程が原盤記録である。そして、この原盤を用いて電鍍プロセスによりスタンプを作製し、このスタンプを用いて射出成型などにより再生専用の光ディスク(レプリカディスク)を量産する。

【0062】一方、光ディスクドライブ装置6は、例えば相変化媒体や光磁気媒体などの記録再生可能な記録媒体を駆動してデータの記録再生を行う装置であり、記録ビット系列に従って半導体レーザを駆動し、この半導体レーザからの光ビームにより光ディスク上にデータの記録を行う。記録されたデータの再生は、半導体レーザからの光ビームを光ディスク上に照射し、その反射光を光検出器で検出することにより行う。

【0063】次に、図10～図11に示すフローチャートを用いて、本実施形態におけるデータ変換の処理手順を説明する。まず、前述した第3のテーブルに記述された代替コードを用いてコード変換を行うかどうかを示す代替コードフラグが立っているか否かを判定する(ステップS11)。ステップS11において代替コードフラグが立っていない場合には、現在の入力データをまず第1または第2のテーブルを用いて主コードAまたは主コードBの通常コードに変換し(ステップS12)、引き続き代替コードフラグをクリアする(ステップS13)。一方、ステップS11において代替コードフラグが立っている場合には、現在の入力データを代替コード



15

に変換し（ステップS14）、引き続いて代替コードフラグをクリアする（ステップS13）。次の入力データは、第1または第2のテーブルを用いて主コードAまたは主コードBの通常コードに変換する（ステップS15）。

【0064】次に、こうしてコード変換処理により得られた連続する2つのコード（現コードと次コード）に対して、両コードの境界部分におけるビット“0”の連続個数RL3を検出し（ステップS16）、この個数RL3を判定する（ステップS17、S19）。ステップS17において、コード境界部分におけるビット“0”の連続個数RL3が $k=8$ 個を越えていると判定された場合には、最後尾ビットを“1”に反転させた反転コードに現コードを変更する（ステップS18）。また、ステップS19においてコード境界部分におけるビット“0”の連続個数RL3が0と判定された場合、つまりコード境界部分においてビット“1”が連続している場合には、現コードを主コードBの特定コードに変更し（ステップS20）、同時に代替コードフラグを立てる（ステップS21）。

【0065】コードの境界条件がこれら以外の場合、つまりコード境界部分におけるビット“0”の連続個数RL3が $k=8$ 個を越えておらず、また0でもない場合には、ビット反転が可能かどうかを判定する（ステップS22）。具体的には、このステップS22では、最後尾ビットを“1”に反転させた反転コードが存在し、かつ、次コードの先頭側におけるビット“0”の連続個数RL1が $d=1$ 以上であれば、ビット反転が可能であり、そうでなければビット反転は不可能であると判定する。

【0066】そして、ビット反転が可能な場合はDSVの絶対値を計算し（ステップS23）、前回の反転可能位置でビット反転した場合の反転系列のDSVと、ビット反転しなかった場合の通常系列のDSVの絶対値を比較し（ステップS24）、反転系列のDSVの絶対値の方が通常系列のDSVの絶対値より小さければ、前回の反転可能位置についてビット反転を行う。すなわち、現コードを反転コードに変更する（ステップS25）。この後ステップS26に移る。ステップS24において反転系列のDSVの絶対値の方が通常系列のDSVの絶対値より大きければ、ビット反転を行うことなくステップS26に移る。

【0067】ステップS26においては、反転可能位置を記憶すると共に、DSVの値を確定させる。この後、DSVの値を更新する（ステップS27）。この更新に際しては、現在の反転可能位置でビット反転した場合の反転系列とビット反転しなかった場合の通常系列のDSVをそれぞれ計算し、ビット反転しなかった場合は、両系列のDSVを更新する。

【0068】最後に、以上のようにして得られたコード

16

とステップS21でセットされた代替コードフラグの出力を行う（ステップS28）。このようにしてビット“0”の連続個数が最小1個、最大8個に制限され、出力ビット系列の直流及び低周波成分が抑圧された $8/12$ 変換によるコード変換を行うことができる。そして、変換されたコードについてビット“1”で出力を反転し、ビット“0”で出力を保持するNRZI変換処理を行って、出力信号系列（記録信号系列）として出力する。

【0069】図12に、本実施形態のデータ変換方法をランダムデータ系列からなる入力データに対して適用した場合の出力信号系列の電力スペクトルを示す。比較のために、同一のデータに対して従来の（1，7）RLL方式を適用した場合の電力スペクトルを図13に示す。

（1，7）RLL方式は、特に直流及び低周波成分の管理を行っていないため、低周波領域のスペクトルはフラットな特性を持っている。それに対して、本実施形態の方式では直流及び低周波成分が小さくなるようにDSVを管理しているため、直流から低周波領域のスペクトルが大幅に抑圧されていることが分かる。

【0070】なお、本発明は上記実施形態に限られるものではない。例えば、コード変換に際して、周期的にビット“1”が偶数個のパターンと奇数個のパターンのいずれかからなる同期コードを挿入するようにしてもよい。その際、同期コードが挿入された位置を前記したビット反転可能位置と同等に扱い、この同期コード直前までのDSVの絶対値が最小となるように、直前の反転可能位置におけるビットを反転させるか否かを判断するようにしてもよい。また、同期コードが挿入された直後の反転可能位置までのDSVの絶対値が最小となるように、同期コードのパターン（ビット“1”が偶数個か奇数個か）を選択するようにしてもよい。

【0071】このように同期コードの挿入位置を反転可能位置と同等に扱って、反転可能位置におけるビット反転や同期パターンの選択を行い、DSVを制御することによって、出力信号系列の直流及び低周波成分をより一層効果的に抑圧することが可能となる。

【0072】

【発明の効果】以上説明したように、本発明によれば $m$ ビットのデータを $n$ ビットのコードに変換する場合に、コード境界部分においてビット“0”の連続個数が所定値よりも大きい場合にはコード最後尾をビット反転させ、このビット反転処理の有無に関わらずビット“0”の連続個数が所定の範囲に含まれる場合には、コードを変換したNRZIパターンのDSVの絶対値がより小さくなるようにビット反転の実行の有無を制御することによって、ビット“0”の連続個数を所定範囲に制限しつつ、出力ビット系列の直流及び低周波成分を抑圧することができる。これにより、本発明を記録媒体への記録ビット系列の生成に用いた場合、光ディスク装置などのデ

ータ記録再生装置におけるトラッキングサーボ等の安定化を図ることができる。

【0073】また、本発明では実施形態で例示したように8/12変換を(1, 8) RLLの条件で行うようにしてクロック周波数が余り高くないようにすることができるため、信号処理回路を安価に実現する上で有利である。

【0074】さらに、出力ビット系列の直流及び低周波成分の抑圧のために調整ビットを挿入する必要がないことから、出力ビット系列を記録媒体への記録ビット系列に用いる場合、実効的な記録容量を損なわないという利点がある。

#### 【図面の簡単な説明】

【図1】本発明の一実施形態に係るデータ変換方法に用いる(1, 8) RLL符号の条件を満たす12ビットパターンの例を示す図

【図2】同実施形態に係るデータ変換方法に用いる主コードAに割り当て可能なビットパターン(第1のテーブル)の例を示す図

【図3】同実施形態に係るデータ変換方法に用いる主コードAの一部とその反転コードを示す図

【図4】同実施形態に係るデータ変換方法に用いる主コードBに割り当て可能なビットパターン(第2のテーブル)の例を示す図

【図5】同実施形態に係るデータ変換方法に用いる主コードBの通常コードと特定コードの対応を示す図

【図6】同実施形態に係るデータ変換方法に用いる代替コードに割り当て可能なビットパターン(第3のテーブル)の例を示す図

【図3】

番号	通常コード	反転コード
1	000100000000	000100000001
2	100100000000	100100000001
3	010100000000	010100000001
4	000010000000	000010000001
5	100010000000	100010000001
6	010010000000	010010000001
7	001010000000	001010000001
8	101010000000	101010000001
9	100001000000	100001000001
10	010001000000	010001000001
11	001001000000	001001000001
12	101001000000	101001000001
13	000101000000	000101000001
14	100101000000	100101000001
15	010101000000	010101000001
16	100000100000	100000100001
17	010000100000	010000100001
18	001000100000	001000100001
19	101000100000	101000100001
20	000100100000	000100100001
21	100100100000	100100100001
22	010100100000	010100100001
23	000010100000	000010100001
24	100010100000	100010100001
25	010010100000	010010100001
26	001010100000	001010100001
27	101010100000	101010100001

\*ル)の例を示す図

【図7】同実施形態におけるコードのビット“1”と“1”との間のビット“0”の連続個数、コード先頭側の“0”の連続個数、コード後尾側の“0”の連続個数およびコード境界部の“0”の連続個数についての説明図

【図8】同実施形態に係るデータ変換方法における通常コードおよび反転コードに対応する出力コード、NRZIパターン、出力波形およびDSVを示す図

【図9】本発明のデータ変換方法を適用したシステムの一例の構成を示すブロック図

【図10】同実施形態に係るデータ変換方法の処理手順を示すフローチャートの一部を示す図

【図11】同実施形態に係るデータ変換方法の処理手順を示すフローチャートの残りの一部を示す図

【図12】同実施形態に係るデータ変換方法により得られる出力ビット系列の電力スペクトルを示す図

【図13】(1, 7) RLL符号を用いたデータ変換方法により得られる出力ビット系列の電力スペクトルを示す図

#### 【符号の説明】

- 1…ROM(第1～第3のテーブル)
- 2…ビット反転器
- 3…NRZI変換器
- 4…制御部
- 5…光ディスク原盤記録装置
- 6…光ディスクドライブ装置

【図5】

番号	通常コード	特定コード	番号	通常コード	特定コード
1	100000010001	10000000101	26	01010001001	010100100101
2	010000010001	01000000101	27	000010001001	000010100101
3	001000010001	00100000101	28	100010001001	100010100101
4	101000010001	10100000101	29	010010001001	010010100101
5	000100010001	00010000101	30	001010001001	001010100101
6	100100010001	10010000101	31	101010001001	101010100101
7	010100010001	01010000101	32	100001001001	10000010101
8	000010010001	00001000101	33	010001001001	01000010101
9	100010010001	10001000101	34	001001001001	00100010101
10	010010010001	01001000101	35	101001001001	10100010101
11	001010010001	00101000101	36	000101001001	000100010101
12	101010010001	10101000101	37	100101001001	100100010101
13	100001010001	10000100101	38	010101001001	010100010101
14	010001010001	01000100101	39	10000101001	100001010101
15	001001010001	00100100101	40	01000101001	010001010101
16	101001010001	10100100101	41	00100101001	001001010101
17	000101010001	00010100101	42	10100101001	101001010101
18	100101010001	10010100101	43	000010101001	00001010101
19	010101010001	01010100101	44	10010101001	100001010101
20	100000010001	100000010101	45	010100101001	010001010101
21	010000010001	010000010101	46	001010101001	001001010101
22	001000010001	001000010101	47	100010101001	101001010101
23	101000010001	101000010101	48	000100101001	000101010101
24	000100001001	000100010101	49	010010101001	010101010101
25	100100001001	100100010101	50	101010101001	100101010101

【図1】

番号	ビットパターン	番号	ビットパターン
1	000100000000	31	100000010000
2	100100000000	32	010000010000
3	010100000000	33	001000010000
4	000010000000	34	101000010000
5	100010000000	35	000100010000
6	010010000000	36	100100010000
7	001010000000	37	010100010000
8	101010000000	38	000010010000
9	000010000000	39	100010010000
10	100001000000	40	010010010000
11	010001000000	41	001010010000
12	001001000000	42	101010010000
13	101001000000	43	000010100000
14	000101000000	44	100010100000
15	100101000000	45	010010100000
16	010101000000	46	001010100000
17	000001000000	47	101010100000
18	100001000000	48	000101010000
19	010001000000	49	100101010000
20	001001000000	50	010101010000
21	101001000000	51	000000010000
22	000100100000	52	100000010000
23	100100100000	53	010000010000
24	010100100000	54	001000010000
25	000010100000	55	101000010000
26	100010100000	56	000100010000
27	010010100000	57	100100010000
28	001010100000	58	010100010000
29	101010100000	59	000010010000
30	000000010000	60	100010010000

番号	ビットパターン
341	100010100101
342	010010100101
343	001010100101
344	101010100101
345	000000010101
346	100000010101
347	010000010101
348	001000010101
349	101000010101
350	000100010101
351	100100010101
352	010100010101
353	000010010101
354	100010010101
355	010010010101
356	001010010101
357	101010010101
358	000001010101
359	100001010101
360	010001010101
361	001001010101
362	101001010101
363	000101010101
364	100101010101
365	010101010101

【図2】

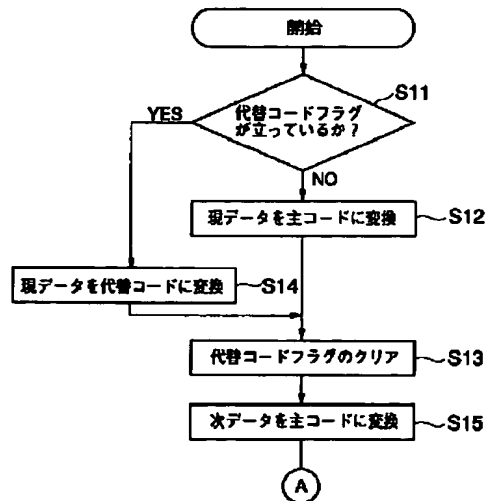
番号	主コードA	番号	主コードA	番号	主コードA
1	000100000000	31	101000010000	181	000100001010
2	100100000000	32	000100001000	182	100100001010
3	010100000000	33	100100001000	183	010100001010
4	000010000000	34	010100001000	184	000010001010
5	100010000000	35	000010010000	185	100010001010
6	010010000000	36	100010010000	186	010010001010
7	001010000000	37	010010010000	187	001010001010
8	101010000000	38	001010010000	188	101010001010
9	100001000000	39	101010010000	189	100001001010
10	010001000000	40	100001010000	190	010001001010
11	001001000000	41	010001010000	191	001001001010
12	101001000000	42	001001010000	192	101001001010
13	000101000000	43	101001010000	193	000101001010
14	100101000000	44	000101010000	194	100101001010
15	010101000000	45	100101010000	195	010101001010
16	100001000000	46	010101010000	196	00000101010
17	010001000000	47	100000010000	197	010000101010
18	001000100000	48	010000010000	198	001000101010
19	101000100000	49	001000010000	199	101000101010
20	000100100000	50	101000010000	200	000100101010
21	100100100000	51	000100001000	201	100100101010
22	010100100000	52	100100001000	202	010100101010
23	000010100000	53	010100001000	203	000010101010
24	100010100000	54	000010001000	204	100010101010
25	010010100000	55	100010001000	205	010010101010
26	001010100000	56	010010001000	206	001010101010
27	101010100000	57	001010001000	207	101010101010
28	100000100000	58	101010001000		
29	010000100000	59	100001001000		
30	001000010000	60	010001001000		

【図6】

番号	代替コード	番号	代替コード
1	000100000000	31	010101010000
2	010100000000	32	000000001000
3	000010000000	33	010000001000
4	010010000000	34	001000001000
5	001010000000	35	000100001000
6	000001000000	36	010100001000
7	010001000000	37	000010001000
8	001001000000	38	010010001000
9	000101000000	39	001010001000
10	010101000000	40	000001001000
11	000000100000	41	010001001000
12	010000100000	42	001001001000
13	001000100000	43	000101001000
14	000100100000	44	010101001000
15	010100100000	45	000000101000
16	000010100000	46	010000101000
17	010010100000	47	001000101000
18	001010100000	48	000100101000
19	000000100000	49	010100101000
20	010000010000	50	000010101000
21	001000010000	51	010010101000
22	000100010000	52	001010101000
23	010100010000	53	010000001000
24	000010010000	54	001000001000
25	010010010000	55	000100001000
26	001010010000	56	010100001000
27	000001010000	57	000010000100
28	010001010000	58	010010000100
29	001001010000	59	001010000100
30	000101010000	60	000001000100

番号	代替コード
111	000010010010
112	010010010010
113	001010010010
114	000001010010
115	010000101010
116	001001010010
117	000101010010
118	010101010010
119	00000001010
120	01000001010
121	001000001010
122	000100001010
123	010100001010
124	000010001010
125	010010001010
126	001010001010
127	000001001010
128	010001001010
129	001001001010
130	000101001010
131	010101001010
132	000000101010
133	010000101010
134	001000101010
135	000100101010
136	010100101010
137	000010101010
138	010010101010
139	001010101010

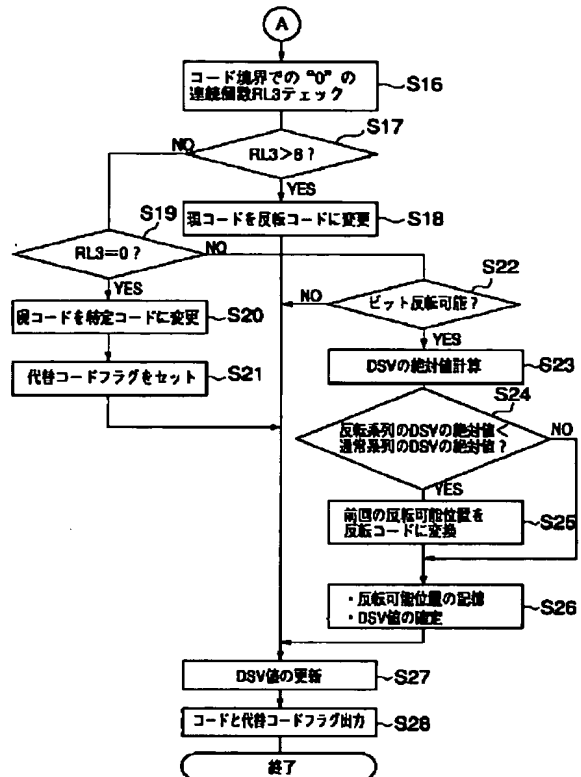
【図10】



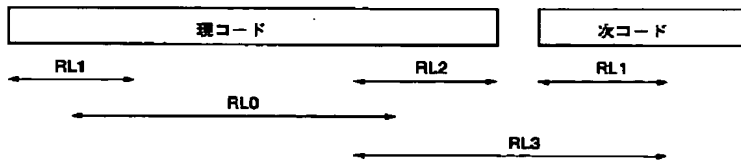
【図4】

番号	主コードB	号	主コードB	番号	主コードB
1	001000000001	36	001001001001	71	010101000101
2	101000000001	37	101001001001	72	1000001000101
3	100000010001	38	000101001001	73	0100001000101
4	010000010001	39	100101001001	74	0010001000101
5	001000010001	40	010101001001	75	1010001000101
6	101000010001	41	100000101001	76	0001001000101
7	000100010001	42	010000101001	77	1001001000101
8	100100010001	43	001000101001	78	0101001000101
9	010100010001	44	101000101001	79	0000101000101
10	000010010001	45	000100101001	80	1000101000101
11	100010010001	46	100100101001	81	0100101000101
12	010010010001	47	010100101001	82	0010101000101
13	001010010001	48	000010101001	83	1010101000101
14	101010010001	49	100010101001	84	100000010101
15	100001010001	50	010010101001	85	010000010101
16	010001010001	51	001010101001	86	001000010101
17	001001010001	52	101010101001	87	101000010101
18	101001010001	53	100000000101	88	000100010101
19	000101010001	54	010000000101	89	100100010101
20	100101010001	55	001000000101	90	010100010101
21	010101010001	56	101000000101	91	000010010101
22	100000000101	57	000100000101	92	100010010101
23	010000000101	58	100100000101	93	010010010101
24	001000000101	59	010100000101	94	001010010101
25	101000000101	60	000010000101	95	101010010101
26	000100000101	61	100010000101	96	100001010101
27	100100000101	62	010010000101	97	010001010101
28	010100000101	63	001010000101	98	001001010101
29	000010000101	64	101010000101	99	101001010101
30	100010000101	65	100001000101	100	000101010101
31	010010000101	66	010001000101	101	100101010101
32	001010000101	67	001001000101	102	010101010101
33	101010000101	68	101001000101		
34	100001001001	69	000101000101		
35	010001001001	70	100101000101		

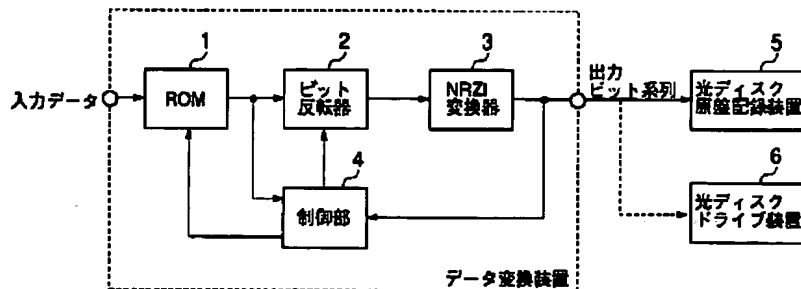
【図11】



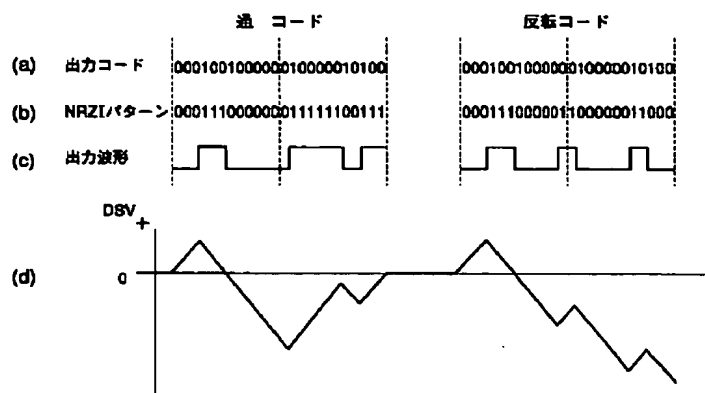
【図7】



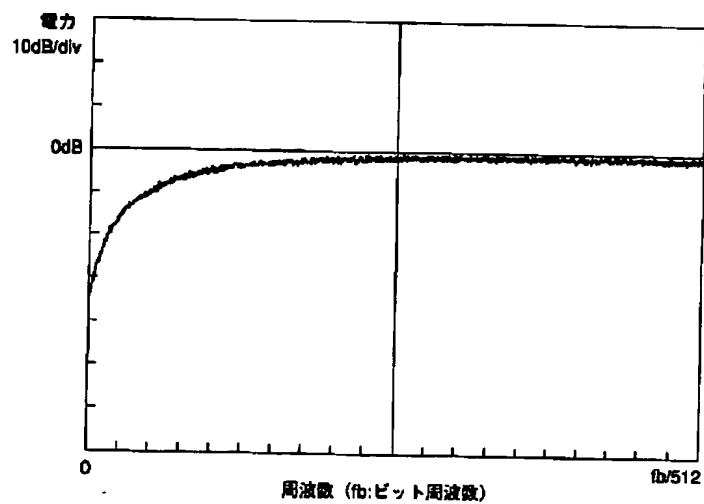
【図9】



【図8】



【図12】



【図13】

